團隊測驗報告

報名序號：109911

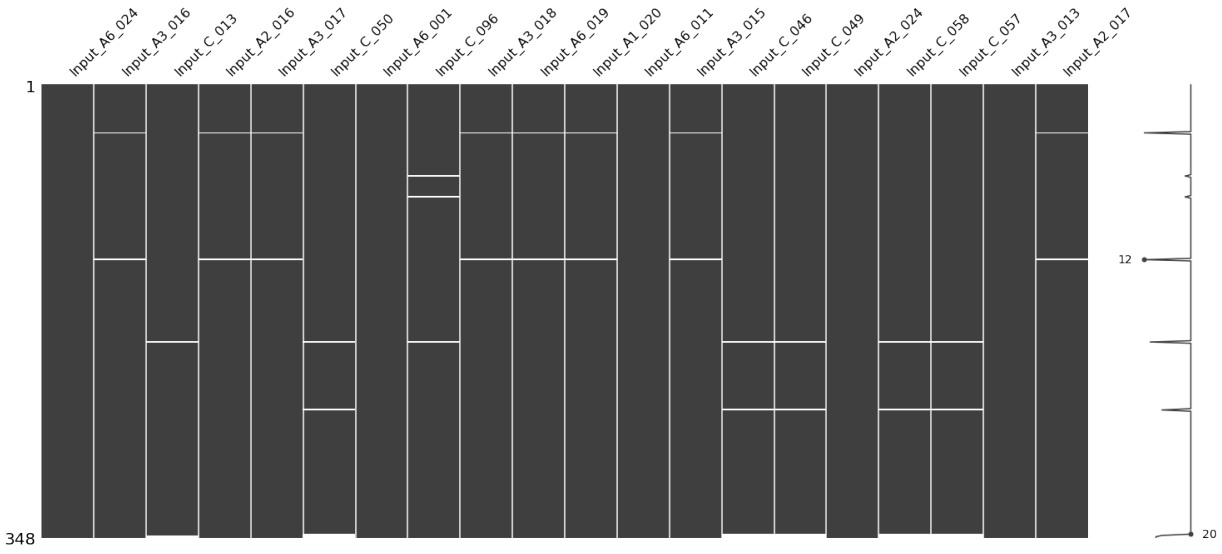
團隊名稱：明顯是個狠角色

註1：請用本PowerPoint 文件撰寫團隊程式說明，**請轉成PDF檔案繳交**。
註2：依據競賽須知第七條，第4項規定：
測試報告之簡報資料不得出現企業、學校系所標誌、提及企業名稱、學校系所、教授姓名及任何可供辨識參賽團隊組織或個人身分的資料或資訊，違者取消參賽資格或由評審會議決議處理方式。

資料前處理

處理Target的缺失值

- 圖中可以看到，白色橫線是在target的20個筆數中為缺失值的位置。為了不影響訓練結果，我們將其丟棄，丟棄9筆target。



偏移值轉換

- Input_C_015~038與Input_C_063~082為偏移量的文字參數。我們將其轉換為x軸的偏移量與y軸的偏移量。舉例來說: Input_C_015中的第一筆資料 N;0;L;1，會變為Input_C_015_x = -1、Input_C_015_y = 0。

N;0;L;1 -> x: -1, y: 0

	Input_C_015_x	Input_C_015_y
0	-1.0	0.0
1	0.0	0.0
2	-1.0	0.0
3	-1.0	0.0
4	-1.0	1.0
...	...	...
334	0.0	0.0
335	0.0	1.0
336	0.0	1.0
337	0.0	0.0
338	0.0	0.0

339 rows × 2 columns

丟棄沒有變異的變數

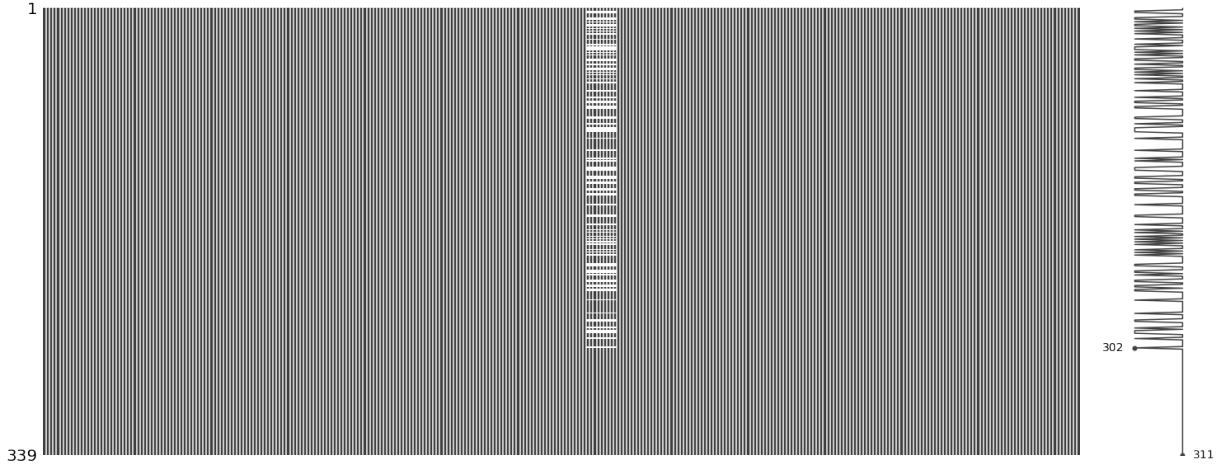
- 在訓練模型的過程中，所有資料都長一樣的變數對於學習沒有任何幫助，甚至會拉低模型的準確度，所以我們也將其丟棄。

```
In [21]: (df.nunique().sort_values() == 1).head(30)

Out[21]: Input_A5_010     True
         Input_C_132      True
         Input_C_006      True
         Input_C_003      True
         Input_C_002      True
         Input_A6_010     True
         Input_A1_010     True
         Input_A2_008     True
         Input_A4_010     True
         Input_A2_010     True
```

標記大量缺失的變數

- 我們發現在Input_C_083~091有大量缺失的變數，我們會新增一個變數(massive_missing)來標記資料在這幾筆中為缺失值。舉例來說，如果Input_C_083為缺失值，massing_missing=1，反之為0。
- 右圖白色橫線為缺失值位置。



偏移量的處理

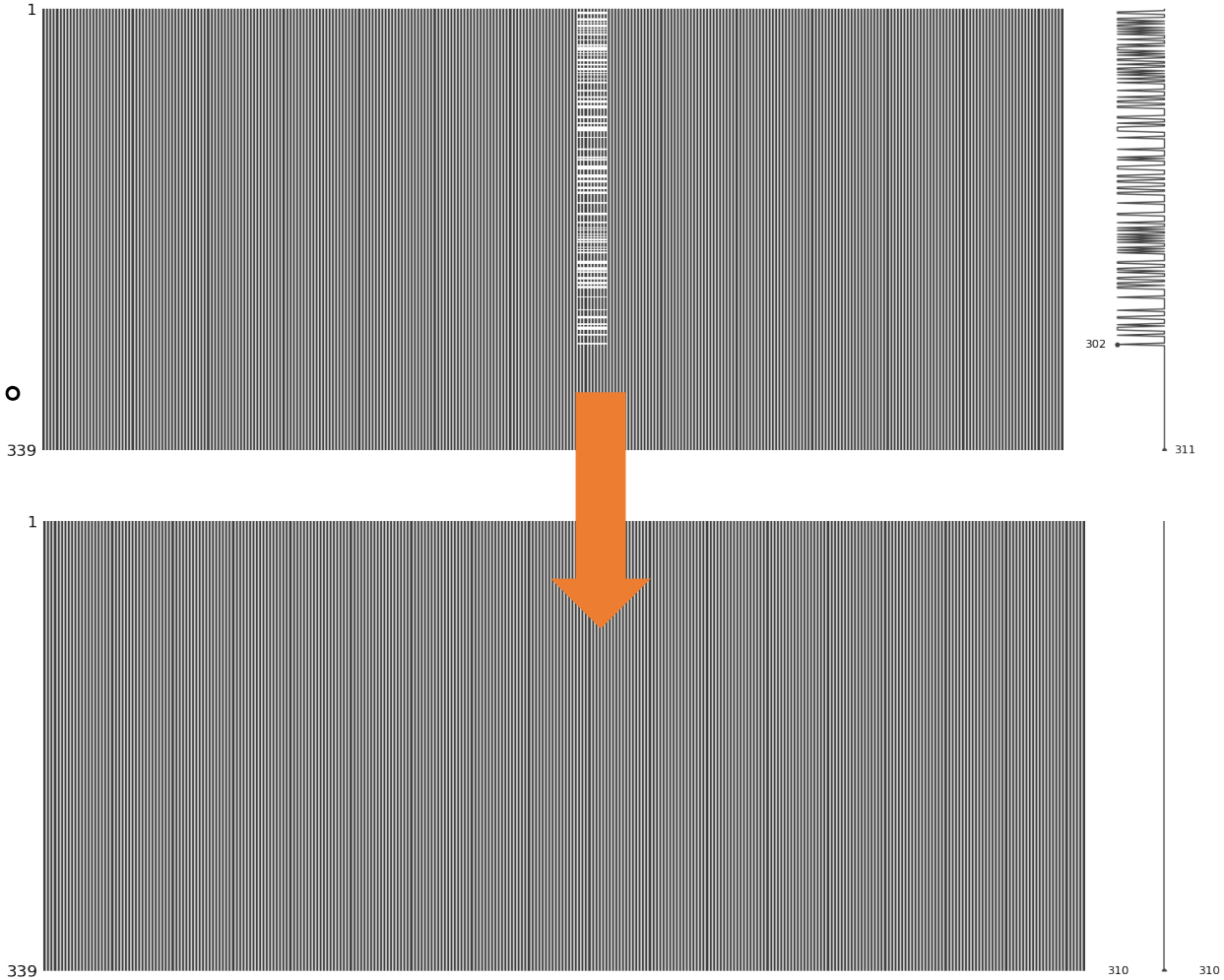
- 我們發現偏移量的部分將其取絕對值會增進模型的準確度。所以將所有Input_C_xxx_x、Input_C_xxx_y都取了絕對值。如右圖所示。

	Input_C_015_x	Input_C_015_y
0	1.0	0.0
1	0.0	0.0
2	1.0	0.0
3	1.0	0.0
4	1.0	1.0
...	...	...
334	0.0	0.0
335	0.0	1.0
336	0.0	1.0
337	0.0	0.0
338	0.0	0.0

339 rows × 2 columns

填補大量缺失的值

- 在前面有說到，Input_C_083~091有大量缺失的變數，我們利用了k-nearest neighbor算法將其缺失的值填滿。



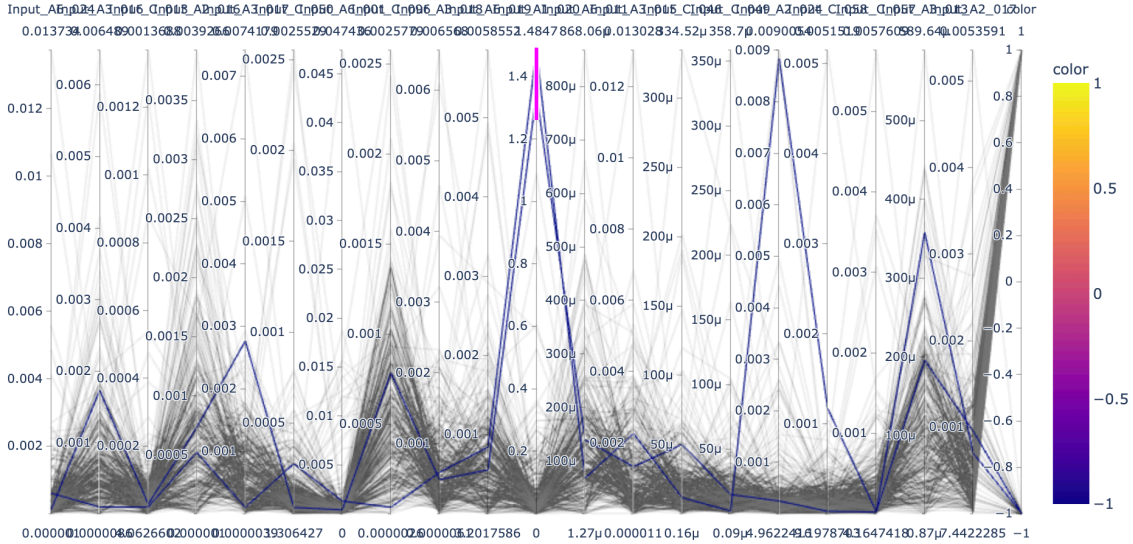
變異過小變數做量化(Quantization)

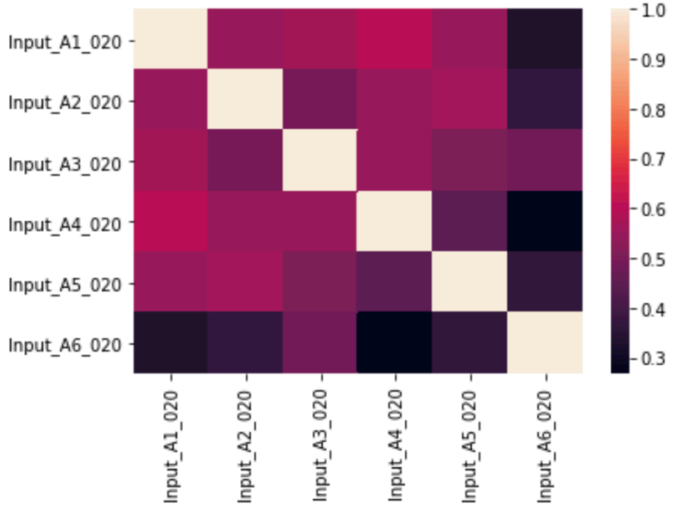
- 針對變異過小的變數，ex: Input_A4_008只有[0.002, 0.004]兩種可能。我們會對其增加one hot encoding的變數。

	Input_A4_008	Input_A4_008_one_hot_0	Input_A4_008_one_hot_1
0	0.002	1.0	0.0
1	0.002	1.0	0.0
2	0.002	1.0	0.0
3	0.002	1.0	0.0
4	0.002	1.0	0.0
...	...	...	...
334	0.002	1.0	0.0
335	0.002	1.0	0.0
336	0.002	1.0	0.0
337	0.002	1.0	0.0
338	0.002	1.0	0.0

Input_A1_020的特徵工程

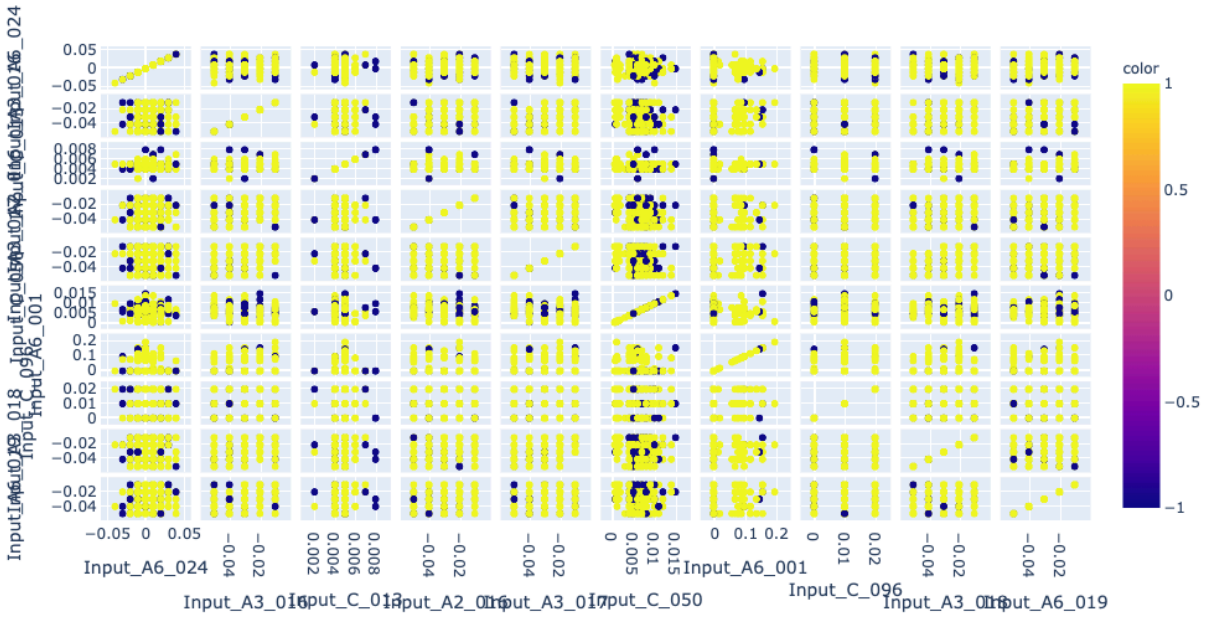
- 在殘差圖中，我們發現Input_A1_020明顯預測的比其他目標變數差。
- 更進一步的我們發現A2~A6的020變數與A1相關性相對較高(從右下相關係數矩陣得知)，於是我們利用了A2~A6的020變數做了一些統計量來當作特徵(min, max, std, mean)。





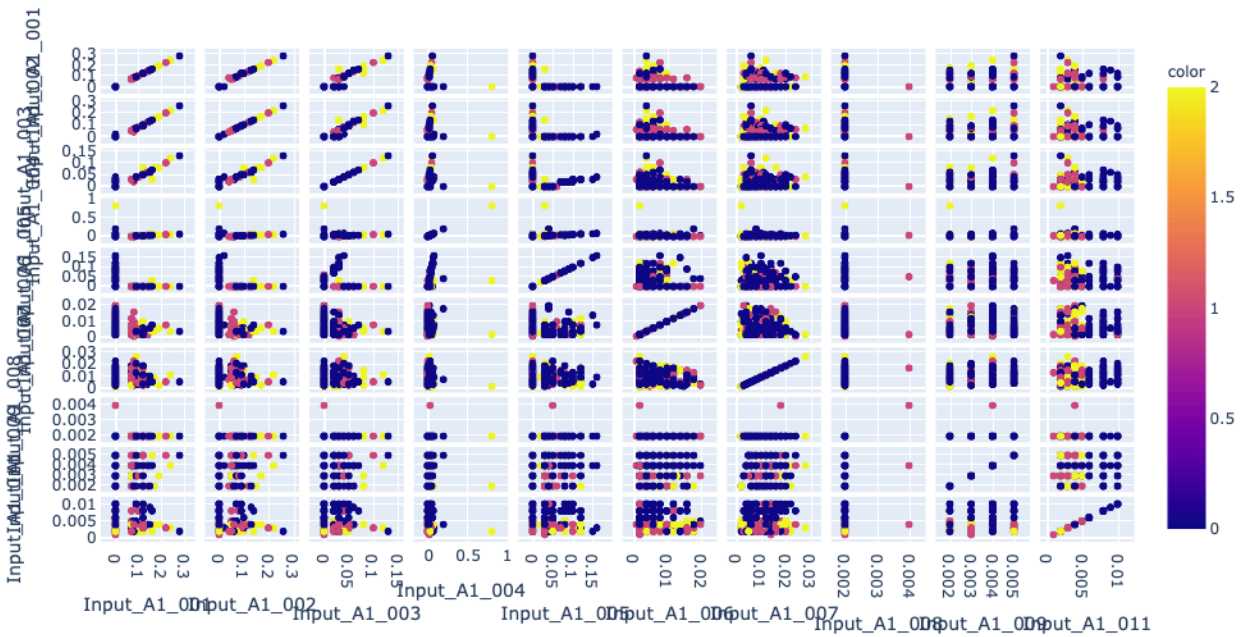
極端值偵測

- 我們發現到資料的outlier會影響到我們的預測結果。
- 偵測極端值的方法為Isolation forest。
- 右圖，我們拿了目標變數的前十個繪製散佈圖，並以顏色區分模型預測的極端值(深色為極端值)。



資料分群

- 我們也發現，將資料分群以後也能增進模型的準確度。
- 右圖為取前十個特徵繪製的散佈圖，並以不同顏色代表不同的群集。



演算法和模型介紹

模型

- 我們使用LightGBM作為我們的模型。
- 在預測20組目標變數時，我們使用了Regression chain的方式來預測。換言之，在預測時，我們會將前一步預測的結果加入模型變成預測下一個變數的特徵之一。
- Regression chain的順序，我們會先預測Input_C系列的目標變數(因為是共同變數)，再預測Input_A系列的變數，因為Input_A1_020是最難預測的，我們會放在最後才預測他。

Hyperparameters Tuning

- 為了預防overfitting與得到更好的準確度，我們對LightGBM的以下參數做了調校，每次調校都做了5-folds的cross validation。
  - max_depth: 每棵樹的最大深度。
  - min_data_in_leaf: 每個leaf中最少需要的資料筆數。
  - subsample: 每次訓練時針對row抽樣比例。
  - colsample_bytree, colsample_bynode: 每次訓練時針對特徵抽樣的比例。
  - num_leaves: leaves的數量。
  - reg_alpha, reg_lambda: 模型的penalty terms。

預測結果

模型調校結果

預測結果